福山大学工学部紀要
第7号　1985年3月

建築教育におけるパーソナルコンピュータ利用の可能性について（２）
（Prolog 言語と述語論理の可能性）

横井友幸*・谷口興紀*

A Note on C.A.I. in the Architectural Education
—Possibility of Prolog language and Predicate logic—

Tomoaki YOKOI and Okinori TANIGUCHI

序

8bitパソコンに較べ大容量のメモリを利用できる16bitパソコンの昨近の普及に伴ない，知識処理に関する情報もパソコンレベルにまで浸透しつつある。知識処理を利用した実用的なパソコンソフトが出現するのも時間の問題であろう。

前稿では，建築教育におけるパソコン利用の位置づけ，問題点および今後の構想等について散述したが，本稿では，建築教育に即したＩＣＡＩ (Intelligent Computer Assisted Instruction)システムを構築するための基礎として，不静定力学の基本的演習問題の単純なＣＡＩプログラミングによるProlog言語の検討および述語論理により組み立てられた設計製図の草案批評反復システムによるEncode型モデルの基本的アイディアについて記す。

１．ＩＣＡＩ 概況

個々の学生に適した学課によく準備された教材を組み込む教育プログラムを構築するＣＡＩの研究は，1960年代初頭より米国を中心に始められ，初期の典型的プログラムは，用意されたテキストを表示するだけの電子頁めくり機的なものや設定された答と学生の答を直すための批評を使って応答する問題演習モニタ的なものであった。

1970年に至り，人工知能（ＡＩ：Artificial Intelligence）の手法をとり入れたＩＣＡＩ研究が起り，教材と教育手順とは独立に表現され，問題と答を直すための批評は学生毎に別々に作成できるようになった。ＣＡＩにおけるＡＩの役割は新しい種々の学習環境を設定可能にすることである。

今日では，既応のＩＣＡＩシステムの主な構成要素は，

1. 専門知識（学生に伝えるもの）
2. 学生モデル（学生の理解程度を示す）
3. 指導戦略（教材を提示する方法）

とシステムと学生との接点となる自然な言語による会話を可能とするインターフェースである，互いに分離されて独立なこれらすべてが，どのシステムでも十分に開発されている訳ではなく，共に主要な研究課題となっている。

代表的な教材は，南米の地理，降雨の因果関係，電気回路，感染症診断，基本的算術演算，ゲームなどである。

筆者らは教育の現状からの必要性に迫られて，ＩＣＡＩシステムの開発に取り組み始めたのであるが，大型計算機ではなく，身近なパソコンを対象とした，前稿において構想したシステムにこれらの既応の技術やＡＩの技法をどこまで取り入れることが出来るかが問題である。

２．Prolog 言語の特徴

知識処理の要である知識表現には，意味ネットワーク，述語論理，プロダクションルールなど種々の方法が提案されている。既応のＩＣＡＩシステムは殆んど人工知能のアセンブラとも呼ばれるＬＩＳＰ言語によって記述されているが，一階述語論理に基づくProlog 言語が，我国では第５世代コンピュータの核言語として採用されたのと相まって，最近注目されている。

＊建築学科，＊建築学科

この Prolog 言語は1973年にフランスで開発された，まだ幼い言語であるが，知識表現の方法の一つとして述語論理が用いられていることを考えれば，その記述に好都合の言語かも知れない。Prolog の特徴を生かしたＩＣＡＩシステムの研究を行うことは新しい発想を得るためには有効であるという考えもある。幸い，パソコン用の Prolog も数種類開発されている。

Prolog では，プログラムの各行はホーン節と呼ばれ次の(1)～(3)のいずれかのように記述される。

(1) 頭部．　　Ｐ．
(2) 頭部：－本体．　即ち　Ｐ：－Q_1,…，Ｑｎ．
(3) ？－本体．　　？－　Q_1,…，Ｑｎ．
　　または：－本体．　　（ｎ≧１）

ここに，Ｐ，Ｑｉ は各々，ＡまたはＡ（Ｂ1，…，Ｂｎ）と書かれ，Ａを述語，Ｂｉを項と呼ぶ。

(1)，(2)は事実と規則あるいは論理的関係を表わし，プログラムの実行は(3)により起動され（処理系に対して問い合わせをする），Q1 からQn の順で各々対応する頭部を捜し，その本体を同様に実行することによりなされる。

以下に Prolog の特徴を列挙する。

○パターンマッチング（項の統合）機能により，他の手続き型言語のように比較のためのアルゴリズムを記述する必要がない。

○この機能により三段論法的推論を行うことができる。すなわち，ある事実と規則の頭部とのマッチングを行い，その本体を実行することにより別の事実を導き出す。

○規則の記述には通常人間が無意識に用いている再帰的呼び出しが可能であり，我々の思考を素直に反映できる。

○再試行を行う。すなわち，一つの結果が得られたとき別の可能性を試みることができる。選択子が排他的で別の可能性がない場合には無駄な再試行はカット述語により除くこともできる。この再試行によって非決定性プログラムを容易に書くことが可能である。

○可逆的述語の記述が可能である。例えば，微分の定義から積分を求めることが述語の定義如何によって可能となる。

◇大域的変数が扱えない。変数の有効域は一つの述語の定義内である。大域的変数を必要とする場合，既に定義した事実（変数値）を書き換えるというような工夫をせざるを得ない。

◇等値性が扱えない。１＋２は論理的には３と等しいが Prolog ではあくまでも１＋２である。従って１＋ｘと２を統合してｘ＝１という解を求めるような芸当は残念ながら出来ない。

◇否定が表現できない。Prolog プログラムに使われる節はホーン節だけであるから論理的に完全な否定は表現できない。Prolog で実現される否定 not(P) は「Ｐが証明できない」という意味である。

◇プログラムデバッグがむつかしい。再試行機能によりエラー箇所がわかりにくくなる。

Prolog は未だ発展途上の言語であり，上記の欠点が改良されれば我々により親しみやすいものとなろう。以上の一般的特徴に対し，パソコン用 Prolog 処理系では，メモリ容量，実行速度，組み込み述語の種類等の問題がある。現存の処理系では大規模なプログラムを入力することはまず不可能であろう。

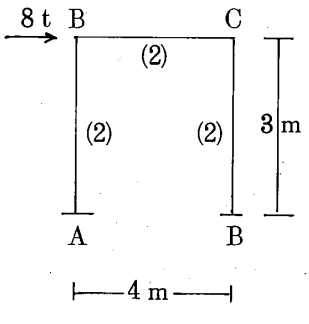
しかしながら，通常簡便さからアルファベットの文字列で表現される述語は，容易に文章に置き換え可能である。このことは，プログラムを読みやすく判りやすいものとする。

３．Prolog 言語によるDecode 型ＣＡＩ モデル

前節では，Prolog 言語の長短について記したが，ここでは，不静定力学における長方形ラーメンのたわみ角法公式適用の演習問題プログラミング（Decode型モデル）を試みることにより，Prolog 言語のＩＣＡＩシステム記述の可能性を検討する。処理系としては，ＮＥＣＰＣ－９８０１Ｆ（メモリ384ＫＢ）用の Prolog －ＫＡＢＡを用いる。

選んだ問題は長方形ラーメン解法の学習における基礎部分であり，初学者は公式適用時にラーメン形状や荷重形等の思い違いや，不注意ミスによる間違いをしばしば犯すところである（図１参照）。この種の間違いはちょっとしたヒントで気付いたり，２～３回の練習により直るものであるが，我々人間が間違いを見つけるときは見落すこともあり，ＣＡＩ向きである。

たわみ角法公式適用における誤答例



○$M_{AB}=2(\varphi_A+\varphi_B+\psi)$
$=2\varphi_B+\underline{\psi}$
○$M_{BA}=2(2\varphi_B+\psi)$
$=4\varphi_B+\underline{\psi}$
○$M_{BC}=2(2\varphi_B+\varphi_C+\psi_{BC})$
$=4\varphi_B+2\varphi_C+\underline{2\psi_{BC}}$
○$M_{AB}=2(\varphi_B)$ ＿
○$M_{BA}=2(2\varphi_B)$ ＿
○$M_{AB}=2\varphi_B\ \underline{-3}$

$$\circ M_{BA} = 4\varphi_B + 3$$
$$\circ M_{AB} = 2(0 + \varphi_B + \psi_{AB}) - 8$$
$$= 2\varphi_B + 2\psi_{AB} - 8$$

図1

プログラムの構成は前稿図5の右側のDecode型モデルに準じ，この検討では特に入力された公式の誤答原因究明に重点に置くことにする。

プログラムは，リストにより記述されたラーメン形状データ，荷重データ，設問データと各解析を行う述語群，それらを呼びだして学習を実行する述語および誤答時のメッセージデータから成る。

この問題では，ラーメン形データ，荷重データを解析し，問題図を表示しながら，形状および節点の状態から考えるべき未知量を分析する。そして，各部材端モーメントを表わすのに必要な公式要素を獲得し，必要に応じて，プログラムエリアへ登録し，更新するという方法が採られる。この時，以後の処理に必要な事実や関係は学習資料として同エリアへ登録しなければならない。これらの事柄は問題データと一緒にdisk上の問題データベースに保存することも考えられるが，誘導可能なものは取り除いて問題をシンプルにした方がdisk効率の上から望ましい。

学生の解答である公式を適用した端モーメント式の形は，項あるいはリストの形式が考えられるが，我々にとって見やすい表現となる項とする。中置記法が使えない場合はリストとした方が扱い易い。実用に供する場合には，入力ミスの起りにくい方法を考える必要があろうが今回の検討では式を項として入力する。また，未知量である節点角，部材角の記号は簡単化のため指定（CRT上で指示する）したものに限定する。

解答の分析は，入力された式（項）を公式の形に合わせてパターンマッチングにより分節化し，問題解析部で得られた各部材公式要素と比較（これもパターンマッチングにより行なわれる）し，判定リストに結果を納めることにより行われる。この判定リストは一種の学生モデルと云えよう。この段階に至るまでに実行が失敗したときは，解答の形式が間違われたとして，2～3通りのメッセージを表示して解答の再入力を促す。正誤判定ならびに誤答箇所の分析は，判定リストを調べることによりなされ，判定リストの要素の値によって種々の誤答原因に対応するメッセージを出力して正答を気づかしめる。

メッセージの出し方が指導戦略につながるが，現段階ではそれは，一つの解答と同時に行なわれ，プログラムと分離してはいない。また，次に節点方程式，層方程式の入力へと進むのが本来の流れであるが，ここでは公式適用までとし，正答に気づけば次の問題へと進める。

Prolog－KABAでは実数値は扱えないので，問題の数値は解が整数値になるように決めておく。ラーメン形状問題データは，問題番号，ラーメン名称，x，y各方向最大長，同CRT画面上での桁数，CRT画面上での図形書出し座標値，同x方向寸法線始点座標，同y方向寸法線始点座標およびラーメンデータのリストであり，ラーメンデータは，書出し節点の記号，支持状態を表わす項，続く部材の方向と向き，長さ，剛比のリスト，次の節点のデータ，続く部材のデータ……のリストである。

Prologには実行の流れを制御する述語が少なく，Basicなどの手続き型言語に浸った者には判りにくいところもあるが，理解が進むと処理の流れを文章に書く通りにプログラミングができることに気づく。述語はすべて独立しており，他のプログラムからも参照できる（F-ortron のサブルーチンと同じ）。従って，制御をうまく工夫すれば，ICAIの三要素を分離することも出来よう。現状では，同タイプの問題は種々扱えても，異種タイプになると解析部と問題パターンが密接に関連しているため，それはむつかしい。

ところで，述語が文章化できることに着目すれば，誤答時の解法説明に述語をデータとして取り出して文章化して示すということも可能である。このようなことは他の言語では到底出来ないことである。

このように，Prologは大きな可能性のある言語であるが，今のところ，実行速度がやや遅く，メモリーを多く必要とするという難点があるが，小規模のシステムの記述には有力であろう。図2～5に，プログラムの出力例とリストの一部を示す。

4．Encode 型モデル

設計製図の授業における草案批評の場面（前回の付録）を述語論理的観点から分析し，そのような過程を，パソコン（MULTI16－BASIC）によって再現するシステムの開発について述べる。

設計製図の授業は，全体としては，Encode 型といえるが，授業の各場面においては，Decode 型といえる面も含んでおり，Encocee型とDecode 型とが，織り混ざっていると考えられる。草案批評の場面の採録を見てみると，学生の応答は，「はい」と「いいえ」とが主であり，矛盾した答えを出したりしている。これらは，教師の言うことを充分にDecode していないからかもしれない。すなわち教師からの一方的な情報発信であって，個人指導の形態を取ってはいても，一般の講義形態で生じている学生の Decode の失敗と軌を一にするといえよ

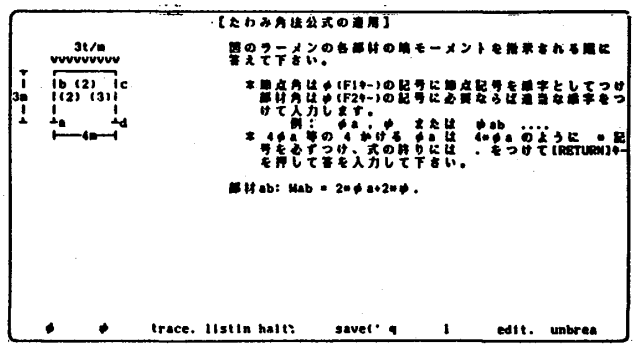


図２ａ　問題表示画面

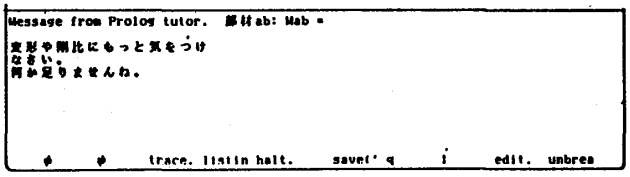


図２ｂ　応答例１

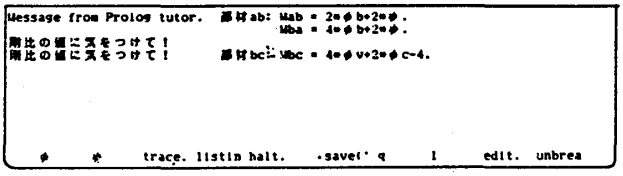


図２ｂ　応答例２

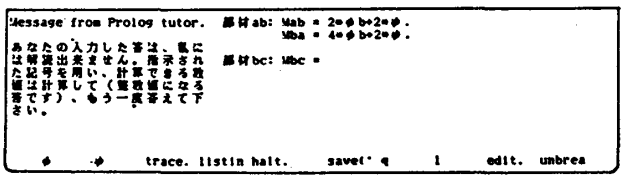


図２ｂ　応答例３

```
/* frame data */

data_t(1.27.1.'【たわみ角法公式の適用】').

data_q(1.29.3.'図のラーメンの各部材の端モーメントを指示される順に').
data_q(1.29.4.'答えて下さい。').

data_n(1.31. 6.'＊節点角はφ(F1キー)の記号に節点記号を添字としてつけ').
data_n(1.33. 7.  '部材角はψ(F2キー)の記号に必要ならば適当な添字をつ').
data_n(1.33. 8.  'けて入力します。').
data_n(1.33. 9.  '  例： φa , ψ または φab ....').
data_n(1.31.10.'＊ 4φa 等の 4 かける φa は 4*φa のように * 記').
data_n(1.33.11.  '号を必ずつけ、式の終りには . をつけて[RETURN]キー').
data_n(1.33.12.  'を押して答を入力して下さい。').

data_ap(1.1.29.'部材').
data_ap(1.2.37).
data_ap(1.3.40).

data_s([1.rahmen.(4.3).(8.4).(6. 9).(6.10).(2.9).
        [jta.s(fix(u))).[u.3.2].j(b.j(2.dr)).[r.4.2].
         j(c.j(2.dl')).[d.3.3].j(d.s(fix(u)))]]).
data_l([1.1].[dist_e.3].[]).
```

図３　問題データ例

```
/*-----------------------------*/
/*   main                      */
/*-----------------------------*/

sdm:-repeat.
       select_thema(N.M.Qstru).
       get_problem(N.M.Qstru.Stru.Load).d_graph.
       problem_analysis(Stru.Jsym.[0.0].Load).
          d_kanji.d_f1.d_f2.prompt(Sp.' = ').
          errormode(Sem.0).d_scroll(Ss.14).
       question_cycle(N).
          prompt(_.Sp).errormode(_.Sem).d_scroll(_.Ss).
    fail.

sdm:-!.write(end).nl.


/* problem analysis */

problem_analysis([j(Jsym.Jstat)|S].Jsym.Pj.Load)
                :-!.
                  count(j.J).
                  make_jvar(Jsym._px).
                  j_var(Jstat.Jvatr).
                  assertz(joint(J.Jsym.Jstat.Pj._px.Jvatr)).
                  draw_j(Jsym.Jstat).
                problem_analysis(S.Jsym.Pj.Load).

problem_analysis([[Dir.Len._k]|S].Jsymp.Pj.[Load|R])
                :-!.
                  count(m.M).
                  slct_type(Dir.Type).
                  get_next_j(S.Jsym).
                  cal_pjn(Jsymp.Dir.Len.Xjn.Yjn).
                  set_jsym(Dir.Jsymp.Jsym.Jx.Jy).
                  cal_load(Load.Len.Cxy.Cyx).
                  assertz(memb(M.Jx.Jy.Type.Len._k.Cxy.Cyx)).
                  draw_m(Type.Dir.Len._k.X.Y.L).
                  draw_l(X.Y.L.Load).
                problem_analysis(S.Jsym.[Xjn.Yjn].R).

problem_analysis([].__.[]):-check_psi.!.
```

図４　list(main およびproblem analysis 一部）

```
/* question cycle */

question_cycle(N):-dsp_q(N).dsp_n(N).
                   sq_cycle(N).

sq_cycle(N):-memb(M.Jx.Jy._._._k.Cxy.Cyx).
             psi(M._q._qa).
             joint(_.Jx._._._px._pxa).
             joint(_.Jy._._._py._pya).
             dsp_memb(N.Jx.Jy).
             response(N.Jx.Jy._k.Cxy._px._pxa._py._pya._q._qa).
             response(N.Jy.Jx._k.Cyx._py._pya._px._pxa._q._qa).
             nl.
            fail.
sq_cycle(_):-!.

response(N.Jx.Jy._k.Cxy._px._pxa._py._pya._q._qa)
                 :-ans_index(N.Jx.Jy).
                   get_ans(N.Sans).
                     ans_analysis(Sans._k.Cxy.
                                    _px._pxa._py._pya._q._qa.Eval).
                     eval(Eval._k.Cxy._px._pxa._py._pya._q._qa).
                   (errflg->fail:true).!.

get_ans(N.Sans):- errorset(read(Sans0).Err).
                  Err==succ.convert(Sans0.Sans).
                 (retract(errflg)->cls_res:true).

get_ans(N.Sans):-!.(errflg->true:err_msg(0)).
                   redo(N).
                 get_ans(N.Sans).

redo(N):-data_ap(N.3.X).d_pos(_.Y).d_cursor(X.Y-1).d_cleol.!.

/* get term & check term */

ans_analysis(A+B._k.Cxy._px._pxa._py._pya._q._qa.Eval)
                  :-!.check_term(B.1._k.Cxy._px._pxa._py._pya._q._qa.Eval).
                  ans_analysis(A._k.Cxy._px._pxa._py._pya._q._qa.Eval).
ans_analysis(A-B._k.Cxy._px._pxa._py._pya._q._qa.Eval)
                  :-!.check_term(B.-1._k.Cxy._px._pxa._py._pya._q._qa.Eval).
                  ans_analysis(A._k.Cxy._px._pxa._py._pya._q._qa.Eval).
ans_analysis(  B._k.Cxy._px._pxa._py._pya._q._qa.Eval)
                  :-!.check_term(B.1._k.Cxy._px._pxa._py._pya._q._qa.Eval).

ans_analysis(_._._._._._._._._._)
                  :-!.err_msg(100).fail.

check_term(C*(A+B)._._._._._._._._._._):-!.err_msg(1).fail.   % ()
check_term(C*(A-B)._._._._._._._._._._):-!.err_msg(11).fail.

check_term(C*U.S._k.Cxy._px._pxa._py._pya._q._qa.Eval)        % 積
                  :-atom(U).!.
                    check_coef(C._k.Mck).
                    check_symbol(U._px._py._q.Mcs).
                    eval1(Mck.Mcs.S._k.Cxy._px._pxa._py._pya._q._qa.Eval).

check_term( T .S._k.Cxy._px._pxa._py._pya._q._qa.Eval)        % 単項－記号
                  :-atom(T).!.
                    check_symbol(T._px._py._q.Mcs).
                    eval2(_k.Mcs.S._k.Cxy._px._pxa._py._pya._q._qa.Eval).

check_term( T .S._k.Cxy._px._pxa._py._pya._q._qa.Eval)        %単項－整数
                  :-integer(T).!.
                    check_const(T.S.Cxy._pxa._pya.Mcl).
                    eval3(Mcl.S._k.Cxy._px._pxa._py._pya._q._qa.Eval).

check_term(_._._._._._._._._._)                               % ERROR
                  :-!.err_msg(100).fail.

/* evaluation */
```

```
eval(Eval._k.Cxy._px._pxa._py._pya._q._qa)
                        :-var_reset(Eval.[_pxa._pya._qa.Cxy]).
                          d_pos(_.Yp).
                          err_eval(Eval).
                          d_cursor(1.Yp).!.

/* ckeck_term sub */

check_coef(2*C._k.['2s'!Mck]):-!.check(C._k.Mck).
check_coef(  C._k.Mck):-atom(C).!.
                        name(C.[X!Y]).
                      ([X]=="k";[X]=="K").Mck=k.
check_coef( _k._k.1):-!.
check_coef(_2k._k.2):-_2k is 2*_k.!.
check_coef(_15k._k.1):-_15k is 3*_k/2.!.
check_coef(_05k._k.2):-_05k is _k/2.!.
check_coef( C ._k.e):-integer(C).!.
check_coef(_._._):-!.err_msg(100).fail.

check_symbol(_px._px._  ._.x):-!.
check_symbol(_py._  ._py._.y):-!.
check_symbol( U ._  ._  .[_qm.Jx.Jy.Jnd].[m.Z.Mz])
                        :-name(_qm.[X.Y]).name(U.[X.Y!Z]),
                          suff(X!Z.Jx.Jy.Mz).Mz¥==e.!.
check_symbol( U ._  ._  .[_  .Jx.Jy._  ].[1.Z.Mz])
                        :-name(U.[X!Z]).(X=:=67;X=:=99).
                          suff(x!Z.Jx.Jy.Mz).Mz¥==e.!.
check_symbol(U._._._.e):-!.err_msg(100.U).fail.

check_const(T.S.Cxy._pxa._pya.Mc1)
                        :-Ts is S*T.C is -Cxy.
                        (Ts=:=Cxy->Mc1=0;Ts=:=C->Mc1=1;Ts=¥=0->Mc1=2).!.
check_const(0.S._.p._.x):-!.
check_const(0.S._._.p.y):-!.
check_const(_._._._._.e):-!.err_msg(100.U).fail.

eval1(2.x.S._k.Cxy._px._pxa._py._pya._q._qa.
        [[Ec._pxa.S._2k._px].._._.._]):-!._2k is 2*_k.
                                        (_pxa=:=0->Ec=[21!Ecr];true).

eval1(1.y.S._k.Cxy._px._pxa._py._pya._q._qa.
        [_.[Ec._pya.S._k._py]._._]):-!.(_pya=:=0->Ec=[21!Ecr];true).

eval1(1.[m.Z.Mz].S._k.Cxy._px._pxa._py._pya._q._qa.
        [_._.[Ec._qa.S._k._q ]._]):-!.(_qa=:=0->Ec=[21!Ecr];true).

eval1(11.x.S._k.Cxy._px._pxa._py._pya._q._qa.
        [[Ec._pxa.S._15k._px]._._.._]):-_15k is 3*_k/2.!.
                                        (_pya==p->true;Ec=[26!Ecr]).

eval1(11.y.S._k.Cxy._px._pxa._py._pya._q._qa.
        [_.[Ec._pya.S._15k._py]._._]):-_15k is 3*_k/2.!.
                                        (_pxa==p->true;Ec=[26!Ecr]).

eval1(12.[m.Z.Mz].S._k.Cxy._px._pxa._py._pya._q._qa.
        [_._.[Ec._qa.S._05k._q]._]):-_05k is _k/2.!.
                                    ((_qa.(_pxa==p;_pya==p))->true:Ec=[26!Ecr]).

eval1(1.x.S._k.Cxy._px._pxa._py._pya._q._qa.
        [[Ec._pxa.S._2k._px].._._.._]):-!.(_pxa=:=1->Ec=[22!Ecr];Ec=[23!Ecr]).
eval1(2.y.S._k.Cxy._px._pxa._py._pya._q._qa.
        [_.[Ec._pya.S._k._py]._._]):-!.(_pya=:=1->Ec=[22!Ecr];Ec=[23!Ecr]).
eval1(2.[m.Z.Mz].S._k.Cxy._px._pxa._py._pya._q._qa.
        [_._.[Ec._pxa.S._k._px]._]):-!.(_pxa=:=1->Ec=[22!Ecr];Ec=[23!Ecr]).
eval1(e.x.S._k.Cxy._px._pxa._py._pya._q._qa.
        [[Ec._pxa.S._2k._px].._._.._]):-!.(_pxa=:=1->Ec=[24!Ecr];Ec=[25!Ecr]).
eval1(e.y.S._k.Cxy._px._pxa._py._pya._q._qa.
        [_.[Ec._pya.S._k._py]._._]):-!.(_pya=:=1->Ec=[24!Ecr];Ec=[25!Ecr]).
eval1(e.[m!M].S._k.Cxy._px._pxa._py._pya._q._qa.
        [_._.[Ec._qa.S._k._q]._]):-!.(_qa=:=1->Ec=[24!Ecr];Ec=[25!Ecr]).

eval1(_._._._._._._._._._._._._):-!.err_msg(100).fail.

eval2(1.y.S._k.Cxy._px._pxa._py._pya._q._qa.
        [_.[Ec._pya.S._k._py]._._]):-!.(_pya=:=0->Ec=[21!Ecr];true).
eval2(1.[m.Z.Mz].S._k.Cxy._px._pxa._py._pya._q._qa.
        [_._.[Ec._qa.S._k._q ]._]):-!.(_qa=:=0->Ec=[21!Ecr];true).
eval2(_.[!.Z.Mz].S._k.Cxy._px._pxa._py._pya._q._qa.
        [_._._.[Ec.Cxy.S]])     :-!.(Cxy=:=0->Ec=[33!Ecr];Ec=[32!Ecr]).
eval2(_._._._._._._._._._._._._):-!.err_msg(100).fail.
```

図5　list(question cycle部)

う。このようなコミュニケーションの阻害を解消する一つの手段として，同じことを繰り返して言う。特に学生の意思によって，同じ批評が繰り返し，好きなだけ再現されるというシステムの開発を構想した。音声情報は，簡便なテープレコーダーで再現可能であるが，図形情報については，ビデオの利用が考えられるが，図形情報の制御と限定（余分な事柄の情報をカットする）という点では，パソコンの方がより適切である。

4－1　草案批評再現システムの構想

草案批評は，教師による一種の設計行為であるので，設計行為そのものの分析が先立たねばならない。しかしその詳細は，別の機会にゆずり，ここでは，前回述べた

(1) 漠然とした全体の要素への分節化
(2) 分節化された要素同士を関係付けること
(3) 分節化された要素の再構成されたものを，再び全体的なものと認知し直すこと。

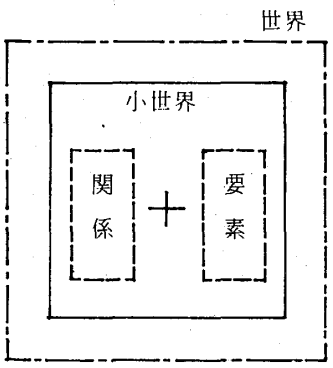
という三つの相の連結によって成り立つと規定する。

Ⅰ）描画部

草案批評の過程において，教師（設計者）は，次の四つのシーンにいるとする。

ⅰ）世界（WORLD）
ⅱ）小世界（MICRO　WORLD）
ⅲ）関係（PREDICATE）
ⅳ）要素（INDIVIDUAL）

これらのシーンの関係は，図－6のように含み含まれ



図－6

るという関係であり，各シーンについて語る（または，考える）ということは，そのシーンの外に出てはじめて可能であるとする。これは，対象言語とメタ言語の関係に似ている。また，要素は，設計製図の場面であるから，室・階段・廊下等であり，全体的なものとは，それらが空間的位置関係をもった一階平面図・二階平面図等である。関係とは，要素について操作 ― 移動・回転・線対称・拡大／縮少等 ― を行うことができることを意味する。小世界とは，当面限定された話しの世界であり，要素と関係より成り立つ。世界とは，小世界を対象として働きかけることができることを意味する。

Ⅱ）再現部

どんな要素が，画面（CRT）に描かれて，それに対してどのような関係付けがなされたかを時間順序で記録しそれを再現する。

4－2　草案批評再現システムの実現

前回の草案批評の対話の内，教師の発言内容を，仮想的CRTを想定し，紙上で分析したものが，表－1である。この表を手懸りに，4－1の構想を実現するシステムのフローチャートを図－7に示す。

4－3　草案批評再現システムの展開

このシステムは，CADシステムとも考えられる。したがって関係の充実により，たとえば，窓情報を与える

ことにより，昼光率の計算とその平面図への表示や，構造計算の組み込みにより，設計の質の向上に役を立てうる。また，学生が草案の作成の段階で使用することにより，学生の設計思考活動の記録を，今まで以上に詳細に採ることができ，設計製図授業の教材や形態の再編のための資料を提供することができよう。ただ，そのためには，パソコンの使用が，極めて容易であり，パソコンを使用していることを忘れている程でなければならないだろう。

現時点ででき上っている部分のＣＲＴ画面を示したものが写真１～４である。

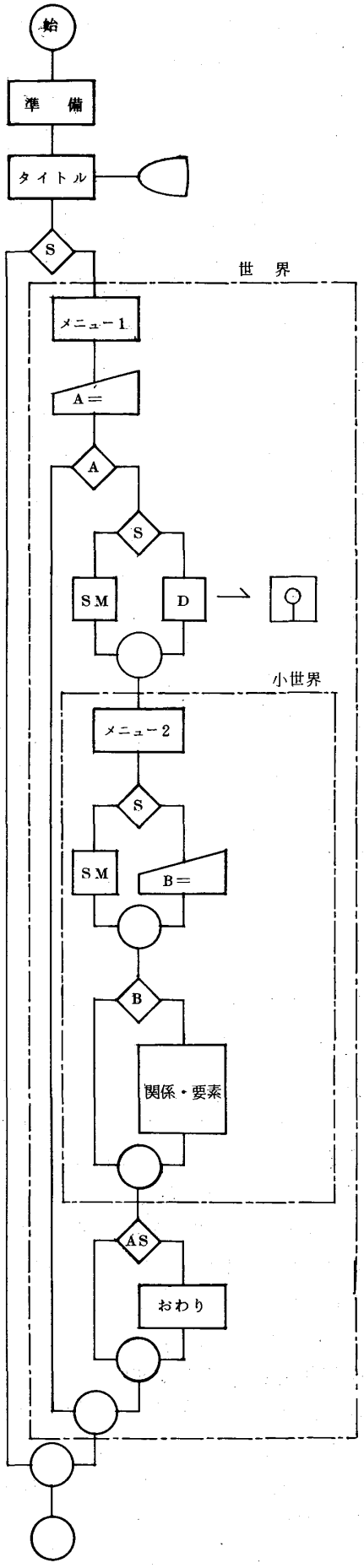
写真１：設計する，または，建築する，ということの展開する世界を象徴的にあらわしたものであり，設計する場合の基準点ともいうべき要素とそれらの関係が与えられており，一つの小世界を構成しており，それを視る我を意識に登場させることにより，常に，既に，世界を構成していることをあらわす。

写真２：与えられた，漠然とした全体ともいうべき赤い平面図から，一つの要素（階段）に注視したことを，黄線にあらわす。

写真３：注視された階段を，右上方位置に移動した図である。

写真４：草案（白線）に種々の要素（形）を描き加え草案を練っている様子をあらわす。

表－１の作成と分析の一部は，技術助手藤井純子氏による。



図－７

表－1　草案批評分析表

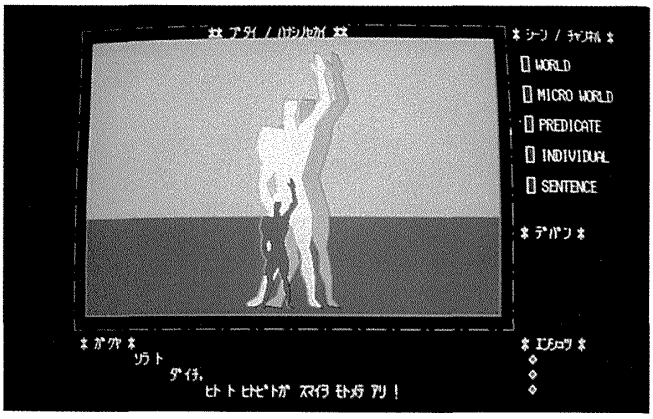
◇：これ(赤)　□：要素化(黄)　△：操作　▽：　○：消去

一階平面図 (1)	1：デジ ◇1 □2 3 4 5 6 7 8
カウンター(2)，喫煙者閲覧室(3)，事務室(4)，個室閲覧室(5)，レファレンス(6)，ラウンジ(7)，ロッカー(8)，－－。	
二階に上ると（二階平面図(1)に話しの世界を移す）。	1：デジ ◇1 □2 3
ここ(2)を通っていかなければ，この階段(3)に行けないのとちがうか何故か。	
通らなくても行けるという方が良いとは思わないか。	
通って行った方が良い（と君は思うのか）？	
（この階段を使うのにこの部屋を通って）行かなくても良いね。	
何故かというと，たとえば，ここ(4)を閉め切ってこの階段(5)を使おうとしてもこの図面(6)のようではできない。	□4 △5 6 7
もし独立に廊下(8)ができていれば，また，ここ(9)を仕切って，ここ(10)を開けることができれば（そういうことはない）ということか。	△8 △9 △10
（と問題点を解消する代案を提示する）。	
この階段(11)は事務職員専用ということかな。	□11
学生は使わないかな。	
ここ(12)も同じだな。	□12
こう廻へ行くと小さな部屋があるというより，こちら側(15)にあると何かと都合が良いのとちがうかな。	△13 14 15
ここ(16)は（何の部屋かな）。	□16
ここ(17)とここ(18)とは閲覧でも性格が違うだろう。	□17 18
ここ(19)は，ここ(20)を通ってこうら(21)に行く。	△19 △20 △21
もしここ(22)からだったら両方(23・24)に対して管理というか，サービスができる。	△22 △23 △24
開架閲覧室(25)といえば，本の書架，書棚をどうならべるかによって管理・サービスの問題も出てくるな。	□25
どう並べる（と考えている）か。	
真中に書架を並べ(26)そうすると壁(27)を作ることになるね。	△26 △27
または，こうならべていって，閲覧席(28)をもうけるか。	△28 P1 △29 P2
学生の案の批評と代案を提示することによって，多くの代替の作成とその中から最上のものを選択することを期待する）	
この上(30)は何になるのか。	△30
自習室(31)というのは騒がしいか静かか。	□31
まあそれだけでもまとまり(32)をもつ。	△32 P3
もう一つは，これ(33)をここにもって来たら(34)，これだけが閲覧・書庫・開架書架として大きなまとまり，小さなまとまり，小さなまとまり，小さなまとまりとなるだろう(35)。	□33 △34 △35 P3
平面図(36)としては，秩序だっているように思わないか。	▽36
こうなってこういう閲覧室(37)だというよりは，物を配置するにしても何をするにしても，照明器具(38)一つにしても，たとえば，	□37 ▽38 ▽39
この部分(40)だけを一つの単位として考えれば良い。	□40
ということは，ここ(1)だけか。	□1
いや各階の平面図(2)においてもいえるかもしれないね。	2：ナマエ
上の階(3)にいったらこっちに行く(4)のか。	□3 △4
じゃあ，もしこれ(5)がここに来て，これ(6)がここに来たらいい訳か(7)(8)。	□5 △6 7 △8
このカウンター(9)は，真中に来て，そこに座わっている人(10)が，こう見まわせば(11)良いという訳か。	□9 △10 △11
というようにやりながら，各階(12)が同じ平面パターンになれば，更に良くなる。	□12
各用途は部を出して来たのだからその用途をどのように編成していくか，まとまるものはまとめてゆくということがこれからの作業方針ということだ。	
トレーシングペーパーに書いて各階を重ね合わせる(13)ということによってやってごらん。	△13
ただし（ ）の中は，状況や暗黙の了解等の説明であり，この批評の所要時間は，8分23秒である。	

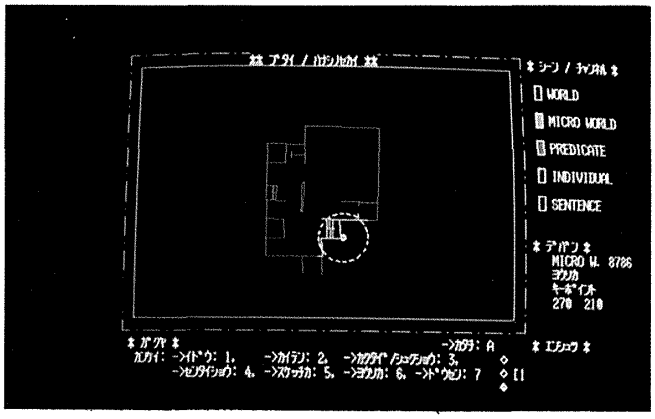
△1：線を描く　△2：長方形を描く　△3：線を消す(陰線を描く)　△4：動線を描く　△5：大円　△6：点　△7：扇形
△8：長方形ペイント　△9：カーソル大円　△10：移動
△P1：長方形とその移動　△P2：□－△－△　△P3：縮小＋α

参考文献

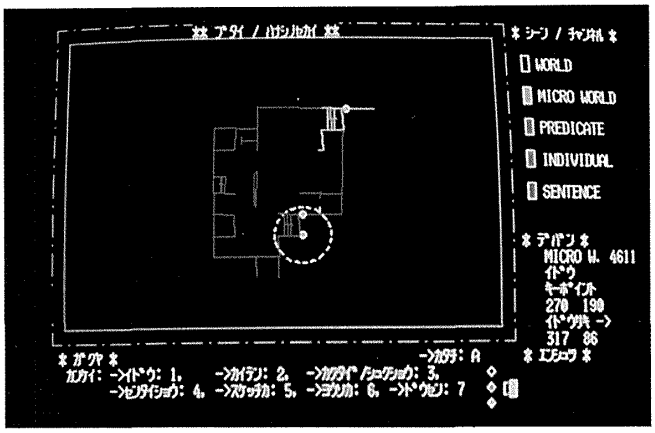
1. A. Barr/E. A. Feigenbaum編，田中幸吉／淵一博監訳，人工知能ハンドブック第Ⅱ巻PP301–398 共立出版
2. Computer Today No.5 1985／1 PP32–38 サイエンス社
3. 中島秀之，Prolog 産業図書
4. 大細孝他3名，Prolog入門 啓学社



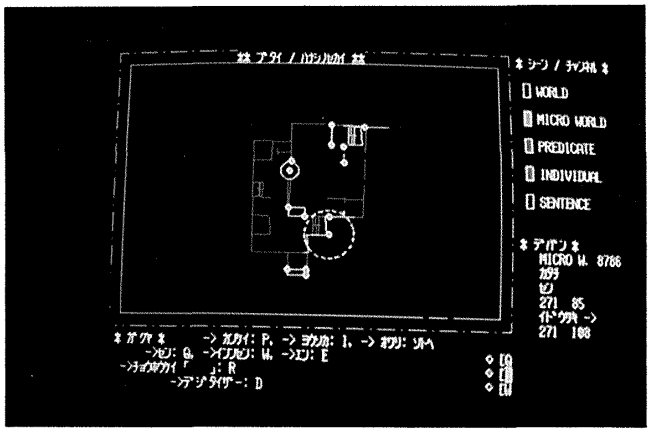
写真－1



写真－2



写真－3



写真－4